Programación de Microservicios con Spring Boot y Red Hat SSO

Nivel Avanzado

Instructor: Carlos Carreño
ccarrenovi@gmail.com

Modulo 10 Manejo de sesiones, expiraciones y refresh tokens

Objetivo: Introducción a Red Hat SSO y su integración con el desarrollo de aplicaciones con Spring Boot

Duración: 1h

Configuración del tiempo de vida de tokens en RH-SSO

1. Ingresa al Admin Console de Keycloak/RH-SSO.
2. Selecciona el **Realm** deseado.
3. Ve a la sección:

"Clients" → Selecciona tu cliente (por ejemplo, spring-boot-client) → pestaña "Settings".

4. Ajusta los siguientes parámetros clave:

continuación

- Parámetros clave

Parámetro	Significado	Ejemplo recomendado
Access Token Lifespan	Tiempo de vida del Access Token (JWT)	`60s` o `5m`
Refresh Token Lifespan	Tiempo de vida del Refresh Token	`30m`, `1h`, `8h`
Client Session Idle	Tiempo inactivo de sesión de cliente	`30m`
Client Session Max	Tiempo máximo total de sesión de cliente	`8h`

Uso de refresh tokens desde aplicaciones cliente

¿Que es un refresh token?

- ✓Es un token largo que permite obtener un nuevo Access Token (JWT) sin necesidad de que el usuario vuelva a autenticarse.
- ✓Sólo debe ser usado por el cliente (SPA, app móvil, Postman, etc.), nunca por los microservicios backend.

Ejemplo: Refresh Token con Postman

1. Pide un token con Authorization Code o Password Flow.
2. Guarda el refresh_token.
3. Realiza el POST anterior para renovar.
4. Usa el nuevo access_token para tus peticiones API.

Manejo de errores por expiración en Spring Boot

- Cuando el Access Token ha expirado, Spring Security detecta automáticamente el **claim exp** y rechaza la petición con:

```
HTTP 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token", error_description="The token is expired"
```